Website prüfen mit Validome

Champs et domaines de saisie

- Définir des champs de saisie à une ligne
- Texte par défaut proposé pour les champs de saisie à une ligne
- Champs de saisie pour mot de passe
- Définir des champs de saisie à plusieurs lignes
- Texte par défaut proposé pour les champs de saisie à plusieurs lignes
- Contrôler le passage à la ligne dans les champs de saisie à plusieurs lignes
- Mettre des champs de saisie en "lecture seule"
- Autres informations

Définir des champs de saisie à une ligne

Les champs de saisie à une ligne servent à l'enregistrement d'un ou de quelques mots ou d'un nombre.

Exemple:

Exemple d'affichage: aperçu

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Strict//EN"
       "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Texte du titre</title>
</head>
<body>

<h1>Formulaire pour entrer le nom</h1>

<form action="input_text.htm">
<p>Prénom:<br>
<input name="prenom" type="text" size="30" maxlength="30">
</p>
<p>Nom de famille:<br>
<input name="nom" type="text" size="30" maxlength="40">
</p>
</form>

</body>
</html>
```

Explication:

`<input>` définit un champ de saisie d'une seule ligne (*input* = *saisie*). Pour être complet, il vous faut ajouter la mention `type="text"`.

Chaque champ de saisie doit recevoir un nom d'identification interne et cela au moyen de la mention `name=`. Le nom attribué est nécessaire pour les scripts CGI, pour pouvoir accéder aux données du champ de saisie. Pour les formulaires-courriel, le nom apparaît comme nom de champ. Et même pour l'accès au champ de formulaire avec JavaScript le nom est important. Le nom ne doit pas être trop long et ne peut pas contenir d'espace ou de caractère spécial. Le premier signe doit être une lettre. Ensuite, les chiffres sont également permis. Utilisez comme caractère spécial tout au plus le tiret de soulignement (`_`), le trait d'union (`-`), les deux points (`:`) ou le point (`.`). Par rapport à JavaScript, le nom ne doit même comporter que des lettres, des chiffres et le tiret de soulignement (`_`). Les majuscules et minuscules sont différenciées de la même façon par la plupart des langages de programmation.

De plus vous devez toujours pour les champs de saisie d'une ligne fixer la longueur d'affichage en nombre de caractères avec `size=` ainsi que la longueur interne du champ en caractères avec `maxlength=`. Les deux mentions signifient le nombre de signes (*size* = *taille, maxlength* = *maximal length* = *longueur maximale*). Si la longueur interne du champ en caractères `maxlength=` est plus grande que la longueur d'affichage en caractères `size=` (comme dans le deuxième des exemples ci-dessus), alors, pour des saisies plus longues le texte défilera automatiquement (dans l'exemple à partir du 31ème caractère entré).

Pour positionner proprement les champs de saisie et leurs inscriptions, la mise en œuvre d'un tableau "sans quadrillage" est d'ailleurs recommandée Tableaux.

Exemple:

Exemple d'affichage: aperçu

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
       "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
```

```
<title>Texte du titre</title>
</head>
<body>

<h1>Formulaire pour la saisie du nom</h1>

<form action="input_text_tableau.htm">
<table border="0" cellpadding="0" cellspacing="4">
 <tr>
  <td align="right">Prénom:</td>
  <td><input name="prenom" type="text" size="30" maxlength="30"></td>
 </tr><tr>
  <td align="right">Nom de famille:</td>
  <td><input name="nom" type="text" size="30" maxlength="40"></td>
 </tr>
</table>
</form>

</body>
</html>
```

Explication:

L'exemple montre le même formulaire que dans l'exemple ci-dessus. Cette fois pourtant, l'inscription et le champ de formulaire correspondant se trouvent proprement l'un à côté de l'autre sur une ligne. Les textes d'inscription ont été alignés à droite et paraissent ainsi bien affectés aux champ de saisie.

L'important est que l'élément `form` soit placé hors du tableau ou plutôt que le tableau soit noté comme élément à l'intérieur du formulaire.

Attention:

Le repère `<input>` est ce qu'on appelle un élément autonome. Quand vous travaillez **conformément à XHTML**, vous devez désigner l'élément `input` comme étant vide de contenu. Pour cela notez le repère autonome sous la forme `<input... />`.
Plus d'informations à ce sujet dans le chapitre sur **XHTML et HTML**.

À l'aide de JavaScript, vous pouvez contrôler les données de l'utilisateur avant qu'il n'envoie le formulaire. Ainsi par exemple, vous pouvez exiger que dans un champ un nombre d'une certaine plage de valeurs soit entré etc. Dans cette documentation, vous trouverez un exemple d'application **Vérifier les saisies de formulaire**.

Texte par défaut proposé pour les champs de saisie à une ligne

Vous pouvez proposer un contenu par défaut pour les champs de saisie à une ligne.

Exemple:

Exemple d'affichage: aperçu

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Strict//EN"
       "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Texte du titre</title>
</head>
<body>

<h1>Écrasez le tout simplement...</h1>

<form action="input_text.htm">
<p>Prénom:<br>
<input name="prenom" type="text" size="30" maxlength="30" value="Brigitte">
</p>
<p>Nom de famille:<br>
<input name="nom" type="text" size="30" maxlength="40" value="Spécimen">
</p>
</form>

</body>
</html>
```

Explication:

Avec l'attribut complémentaire `value=` vous pouvez fixer un texte qui occupera le champ par défaut (*value* = *valeur*).

Champs de saisie pour mot de passe

Pour la saisie de codes secrets, de mots de passe etc. il y a un type spécial de champ de saisie. Les caractères saisis y sont représentés par des signes de repérage (le plus souvent des étoiles), de façon à ce que des personnes présentes dans la pièce de l'utilisateur ne puissent pas lire par inadvertance le mot de passe entré .

Exemple:

Exemple d'affichage: aperçu

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Strict//EN"
       "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Texte du titre</title>
</head>
<body>

<h1>Ne laissez pas voir le clavier ... :-)</h1>

<form action="input_password.htm">
<p>mot de passe:<br>
<input name="motpasse" type="password" size="12" maxlength="12">
</p>
</form>

</body>
</html>
```

Explication:

Champs de saisie pour mot de passe sont définis avec `type="password"` dans le repère `<input>`.

Attention:

Les mots de passe sont, malgré la saisie cachée, transmis en clair sur Internet. Attirez l'attention de l'utilisateur sur ce point de façon explicite. Pour une communication cryptée entre le navigateur et le serveur existe le protocole HTTPS que le serveur Web doit soutenir. Demandez le cas échéant à votre fournisseur d'accès ou à votre webmestre.

Un champ pour mot de passe en HTML ne permet pas encore l'appel de pages WWW protégées. Lisez à ce sujet la partie sur .htaccess - contrôler les réactions du serveur.

Définir des champs de saisie à plusieurs lignes

Les champs de saisie à plusieurs lignes servent à l'enregistrement de commentaires, nouvelles etc.

Exemple:

Exemple d'affichage: aperçu

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Strict//EN"
       "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Texte du titre</title>
</head>
<body>

<h1>Spontanément</h1>

<form action="textarea.htm">
<p>Quels éléments HTML vous viennent à l'esprit et quel effet ont-ils:<br>
<textarea name="entree_utilisateur" cols="50" rows="10"></textarea>
</p>
</form>

</body>
</html>
```

Explication:

`<textarea ...>` introduit un champ de saisie à plusieurs lignes (*textarea* = *passage de texte*). Chaque champ de saisie à plusieurs lignes doit recevoir comme un ↑ champ de saisie à une ligne un nom d'identification interne et cela au moyen de la mention `name=`.

La mention de la largeur du domaine de saisie est obligatoire. L'attribut `rows=` détermine le nombre de lignes affichées (*rows* = *lignes*) et donc la hauteur, tandis que `cols=` fixe le nombre de colonnes affichées (*cols* = *columns* = *colonnes*. "colonnes" veut dire ici le nombre de caractères (par ligne).

Avec `</textarea>` vous fermez le champ de saisie à plusieurs lignes. Le repère de fermeture est nécessaire et ne peut pas être omis.

Attention:

Les mentions `rows=` et `cols=` déterminent en fin de compte la taille d'affichage du champ de saisie, et non pas la longueur permise du texte. Elle est théoriquement illimitée. Les navigateurs WWW équipent en général leurs champs de saisie à plusieurs lignes de barres de défilement de façon à ce que l'utilisateur puisse faire défiler ses données si elles sont plus longues.

Texte par défaut proposé pour les champs de saisie à plusieurs lignes

Vous pouvez proposer un contenu par défaut pour les champs de saisie à plusieurs lignes.

Exemple:

Exemple d'affichage: aperçu

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Strict//EN"
       "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Texte du titre</title>
</head>
<body>

<h1>Écrivez une histoire!</h1>

<form action="textarea.htm">
<p>Ici commence l'histoire:<br>
<textarea name="saisie_utilisateur" cols="50" rows="10">
Il faisait sombre, humide et brumeux, un vrai soir de novembre. Gérard avait presque oublié
 ce qui s'était passé et déambulait perdu dans ses pensées et comme vidé sur la
chaussée. Quand soudain...
</textarea></p>
</form>

</body>
</html>
```

Explication:

Pour doter les champs de saisie de plusieurs lignes d'un texte par défaut, notez le texte désiré simplement comme contenu d'élément entre le repère d'ouverture `<textarea>`, et le repère de fermeture `</textarea>`.

Contrôler le passage à la ligne dans les champs de saisie à plusieurs lignes

Ceci ne fait pas partie du standard HTML parce que le consortium W3 part du principe que le texte dans un champ de saisie à plusieurs lignes passe à la ligne automatiquement lors de la saisie. Ce n'est pourtant pas le cas pour Netscape jusqu'à la version 4.x comprise. Là le texte s'inscrit automatiquement à la suite tant que l'utilisateur n'appuie pas sur la touche retour pour commencer une nouvelle ligne. Pour activer l'option du passage à la ligne automatique, Netscape a inauguré dans le temps un nouvel attribut pour assurer le contrôle du passage à la ligne. L'Explorer Internet interprète lui aussi cet attribut. Mais comme c'est le cas pour les éléments de langage spécifiques - est apparue une quantité d'affectations de valeur désordonnée. Et quand vous utilisez l'attribut, vous créez en tous cas du code HTML non valide.

Exemple:

Exemple d'affichage: aperçu

```
<html>
<head>
<title>Contrôler le passage à la ligne dans les champs de saisie à plusieurs lignes</title>
</head>
<body>

<h1>Passez à la ligne, mais pas l'arme à gauche!</h1>

<form action="textarea_wrap.htm">
<p>une fois doux:<br><textarea name="doux" cols="30" rows="3" wrap="soft"></textarea></p>
<p>une fois sec:<br><textarea name="sec" cols="30" rows="3" wrap="hard"></textarea></p>
<p>une fois effectif :<br><textarea name="effectif" cols="30" rows="3" wrap="physical"></textarea></p>
<p>une fois apparent:<br><textarea name="apparent" cols="30" rows="3" wrap="virtual"></textarea></p>
<p>et une fois éteint:<br><textarea name="eteint" cols="30" rows="3" wrap="off"></textarea></p>
</form>
```

```
</body>
</html>
```

Explication:

Avec l'attribut non-standardisé `wrap=` vous pouvez diriger la césure (*wrap* = *césure*). Les mentions suivantes sont "sous réserve":
`wrap="soft"` est interprété par Netscape 4.x et provoque un passage à la ligne automatique à l'affichage. Lors de l'envoi du formulaire pourtant, aucun passage à la ligne aux endroits où ils s'effectuent n'est transmis.
`wrap="hard"` est interprété par Netscape 4.x et provoque un passage à la ligne automatique à l'affichage. Lors de l'envoi du formulaire, les passages à la ligne aux endroits où ils s'effectuent sont transmis.
`wrap="virtual"` est interprété par l'Explorer Internet MS à partir de la version 4.x et par Netscape versions 2.x et 3.x et signifie la même chose que `wrap="soft"`.
`wrap="physical"` est interprété par l'Explorer Internet MS à partir de la version 4.x et par Netscape versions 2.x et 3.x et signifie la même chose que `wrap="hard"`.
`wrap="off"` est interprété par l'Explorer Internet MS à partir de la version 4.x et par Netscape et signifie la même chose que le réglage par défaut de Netscape jusqu'à la version 4.x - à savoir pas de passage automatique à la ligne.

Mettre des champs de saisie en "lecture seule"

Vous pouvez contraindre un "champ de saisie" à ne plus être un champ de saisie mais à proprement parler seulement un champ de sortie. Cela peut être intéressant par exemple si vous désirez afficher des valeurs transmises par JavaScript ou si vous voulez affecter des champs avec une valeur par défaut que l'utilisateur ne doit pas pouvoir modifier.

Exemple:

Exemple d'affichage: aperçu

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Strict//EN"
       "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Texte du titre</title>
</head>
<body>

<h1>Pas de rabais et pas de quartier</h1>

<form action="input_text_value.htm">
<p>Prix:<br>
<input name="prix" type="text" size="8" value="&euro; 699.-" readonly>
</p>
<p>acceptation de licence:<br>
<textarea name="licence" cols="50" rows="10" readonly>
En envoyant ce formulaire, j'accepte cet accord de licence et déclare l'avoir lu etc...
</textarea>
</p>
</form>

</body>
</html>
```

Explication:

Grâce à l'attribut `readonly` vous pouvez mettre un champ de saisie à une ligne ou un passage de saisie à plusieurs lignes en "lecture seule".

Attention:

Chez les navigateurs plus anciens, cette mention ne fonctionne pas et l'utilisateur peut écraser (remplacer) son contenu!

Quand vous voulez travailler conformément à XHTML, vous devez noter l'attribut sous la forme `readonly="readonly"`. Vous en apprendrez plus à ce sujet dans le chapitre XHTML et HTML.

Autres informations

Dans le sommaire de référence HTML vous trouverez des données précisant où les éléments décrits ici peuvent être mis, quels attributs ils peuvent avoir et ce à quoi il faut veiller pour ces différents attributs:
- référence pour les éléments pour les champs de saisie à une ligne (`<input>`)
- référence pour les attributs pour les champs de saisie à une ligne (`<input>`)
- référence pour les éléments pour les champs de saisie à plusieurs lignes (`<textarea>...</textarea>`)
- référence pour les attributs pour les champs de saisie à plusieurs lignes (`<textarea>...</textarea>`)

Listes de choix, cases d'option et cases à cocher
Définir des formulaires

SELFHTML/Aides à la navigation HTML/XHML Formulaires

© 2001 Stefan Münz / © 2003 Traduction Serge François, 13405@free.fr
selfhtml@selfhtml.com.fr